## PREPARING THE REQUIREMENTS SPECIFIC

### Translating Needs to Specifications (3.4.1 p35)

- One need ⇒ one spec
- Must be complete (all needs covered)
- Must be consistent (independent, no contradictions among needs)
- May be an iterative process - have to re-visit / revise Problem Definition
- Rely on experience and expertise, but for help can also:
    1. search out expert sources,
    2. analyze similar designs,
    3. conduct tests or experiments

### Specifying the Interfaces (3.4.2 p36)

- Should describe in specific terms all points of user interaction (operation and user maintenance if applicable)
- Also specifies all other interfaces such as power connections (if external), physical mounting etc.

### Handling Excessive Requirements (3.4.3 p37)

- Myth: added functionality is free if it doesn't increase the parts count. Truth: life cycle costs (incl. design, development, testing, marketing, warranty and support costs) all increase, often exponentially (**Fig 3.11**), even if the parts costs are the same.
- More stringent spec's cost more (for a lot of the same reasons as above *plus* component costs and potentially manufacturing and shipping). This is sometimes, *but not always*, related to increased reliability (**Fig 3.12**). (e.g. being able to set a buoyancy "depth" to a closer limit can easily cost more both in design and components and could actually *decrease* reliability because of more complicated design). Owe it to client to meet but not exceed - question what they are willing to pay for and what the implications are for reliability and cost.

### Verification of Solution (3.4.4 p38)

- Preliminary Test Plan - how you plan to prove that solution works and spec's are met. (system or acceptance tests come later, but this is the start…)
    - some things can't be tested directly (e.g. failure rate, long term stability)
    - reference to documentation, standards
    - procedures
    - pass / fail limits where appropriate
- consideration of testing starts up front (needs assessment) and continues throughout design (including future maintenance/repair issues)
- regarding spec's - "if you can't prove it, don't say it" - helps refine your spec's to something quantifiable / measurable.
- As with other areas of product development, the later problems are found the more expensive it is to correct. (**Fig 6.4**), but better to find them through system testing than in the field.

**Documentation (3.4.5 p38, Table 3.5 p39)**

- (overview / background) What are we doing and why? (previously covered)
- Problem statement. Previously covered. Can include a brief summary of the problem statement in the actual Requirements Spec section- depends on size & circumstance. (probably not necessary in this case)
- Actual specification - thorough description of the problem in quantified, technical terms. Can be grouped in “technical thinking” terms - targeted at designer, technical people.
- Preliminary Test Plan - (see verification) describes how you will be able to prove that the system works (needs have been met)
- Deliverables - an expansion of the brief discussion included in the Problem statement - again stated in technical terms.
- This is a good time to include source info and standards info etc. in an appendix.
- Other considerations - often looking ahead to issues that may have to be resolved regarding manufacture, maintenance, warranty etc.

**Example: Appendix A.3 - Case Study Requirements Spec. (p134)**